IN THE CLAIMS

Please amend the claims as follows:

1. (Currently Amended) A method for achieving crypto-syncronization in a packet data communication system, the packet data communication system comprising a transmitter and a receiver, said transmitter and said receiver each having cryptographic security capabilities, comprising ~~the steps of~~:

generating data frames at a predetermined rate in a transmitter;

incrementing a state vector at said predetermined rate;

providing said state vector to an encryption module;

generating a codebook from said encryption module, using at least said state vector, said codebook for encrypting at least one of said data frames; ~~and~~

dropping one or more of said frames; and

disabling said state vector from incrementing for each ~~when one or more~~ of said data frames ~~are~~ being dropped.

2. (Original) The method of claim 1 wherein said state vector is enabled after a desired number of said data frames have been dropped.

3. (Currently Amended) The method of claim 1 wherein said ~~the step of~~ generating said data frames comprises ~~the steps of~~:

converting information into digitized information ~~a digital format~~;

providing said digitized information to a vocoder; and

generating said data frames by said vocoder at said predetermined ~~first~~ rate.

4. (Currently Amended) The method of claim 1 wherein said ~~the step of~~ dropping one or more of said data frames comprises ~~the step of~~ dropping said data frames at a fixed, predetermined rate.

5. (Currently Amended) The method of claim 1 wherein said ~~the step of~~ dropping one or more of said data frames comprises ~~the steps of~~:

determining a communication channel latency; and

dropping said data frames at a variable rate in accordance with said communication channel latency.

6. (Currently Amended) The method of claim 5 wherein said ~~the step of~~ dropping said data frames at a variable rate comprises ~~the steps of~~:

decreasing said rate if said communication channel latency falls below at least one predetermined threshold; and

increasing said rate if said communication channel latency exceeds at least one other predetermined threshold.

A8

7. (Currently Amended) The method of claim 1 wherein said ~~the step of~~ dropping said data frames comprises ~~the steps of~~:

determining a communication channel latency;

dropping said data frames at a first predetermined fixed rate if said communication channel latency falls below a predetermined threshold; and

dropping said data frames at a second predetermined fixed rate if said communication channel latency exceeds said predetermined threshold.

8. (Currently Amended) The method of claim 1 wherein said ~~the step of~~ dropping one or more of said data frames comprises: ~~the steps of;~~

determining a communication channel latency; and

dropping each of said data frames having an encoded rate equal to a first encoding rate if said communication channel latency exceeds a predetermined threshold.

9. (Currently Amended) The method of claim 8, further comprising ~~the step of~~ dropping each of said data frames having an encoded rate equal to said first encoding rate and a second encoding rate if said communication channel latency exceeds a second predetermined threshold.

10. (Currently Amended) A method for achieving crypto-syncronization in a packet data communication system, the packet data communication system comprising a transmitter and a receiver, said transmitter and said receiver each having cryptographic security capabilities, comprising ~~the steps of~~:

~~generating~~ receiving data frames at a receiver;

storing said data frames in sequence in a queue;

providing said stored data frames, in sequence, to a decryption module;

~~dropping one or more of said data frames in said queue;~~

incrementing a state vector at a predetermined rate;

providing said state vector to [[a]] the decryption module;

generating a codebook from said decryption module, using at least said state vector, said codebook for decrypting at least one of said data frames; ~~and~~

dropping one or more of said data frames in said queue; and

adjusting said state vector for each of said one or more data frames that are dropped.

11. (Currently Amended) The method of claim 10 wherein said ~~the step of~~ adjusting said state vector comprises ~~the steps of~~:

determining a number of dropped data frames; and

advancing said state vector in proportion to said number of dropped frames.

12. (Currently Amended) The method of claim 11 wherein said ~~the step of~~ advancing said state vector comprises ~~the step of~~ advancing said state vector by a value of one for each of said one or more dropped frames.

a2

13. (Currently Amended) The method of claim 10 further comprising ~~the steps of~~:

applying said adjusted state vector to said decryption module;

generating a second codebook derived from said adjusted state vector;

providing a sequential non-dropped frame in said queue to said decryption module; and

decrypting said sequential non-dropped frame using said second codebook.

14 (Currently Amended) The method of claim 10 wherein said ~~the step of~~ dropping one or more of said data frames comprises ~~the step of~~ dropping said one or more data frames at a fixed rate.

15. (Currently Amended) The method of claim 10 wherein said ~~the step of~~ dropping one or more of said data frames comprises ~~the steps of~~:

determining a communication channel latency; and

dropping said one or more data frames at a variable rate in accordance with said communication channel latency.

16. (Currently Amended) The method of claim 15 wherein said ~~the step of~~ dropping said one or more ~~or~~ of said data frames at a variable rate comprises ~~the steps of~~:

decreasing said rate if said communication channel latency falls below at least one predetermined threshold; and

increasing said rate if said communication channel latency exceeds at least one other predetermined threshold.

17. (Currently Amended) The method of claim 10 wherein said ~~the step of~~ dropping said one or more of said data frames comprises ~~the steps of~~:

determining a communication channel latency;

dropping said data frames at a first predetermined fixed rate if said communication channel latency falls below a predetermined threshold; and

dropping said data frames at a second predetermined fixed rate if said communication channel latency exceeds said predetermined threshold.

18. (Currently Amended) The method of claim 10 wherein said ~~the step of~~ dropping one or more of said data frames comprises: ~~the steps of;~~

determining a communication channel latency; and

dropping each of said data frames having an encoded rate equal to a first encoding rate if said communication channel latency exceeds a predetermined threshold.

19. (Currently Amended) The method of claim 18, further comprising ~~the step of~~ dropping one or more of said data frames having an encoded rate equal to said first encoding rate and a second encoding rate if said communication channel latency exceeds a second predetermined threshold.

20. (Currently Amended) A method for achieving crypto-syncronization in a packet data communication system, the packet data communication system comprising a transmitter and a receiver, said transmitter and said receiver each having cryptographic security capabilities, comprising ~~the steps of~~:

~~generating~~ receiving data frames at a receiver;

storing said data frames in a queue;

providing at least one of said data frames from said queue to a decryption module if available in said queue;

providing a state vector to said decryption module, said state vector incremented at a predetermined rate;

generating a codebook from said decryption module, using at least said state vector, said codebook for decrypting at least one of said data frames; and

disabling said state vector when said queue is in an underflow condition.

21. (Currently Amended) The method of claim 20, wherein said ~~the step of~~ disabling said state vector comprises ~~the steps of~~:

determining that none of said data frames are available for decryption in said queue;

disabling said state vector;

determining that at least one of said data frames is available for decryption in said queue;

enabling said state vector; and

incrementing said state vector by a value of one.

22. (Original) A transmitter for achieving crypto-syncronization in a packet data communication system, the packet data communication system comprising said transmitter and a receiver, said transmitter and said receiver each having cryptographic security capabilities, said transmitter comprising:

means for generating data frames at a predetermined rate;

means for generating a state vector, said state vector incremented at said predetermined rate;

an encryption module for generating a codebook from at least said state vector, said codebook for encrypting at least one of said data frames; and

a processor for dropping one or more of said data frames and for disabling said state vector for each of said data frames that are dropped.

23. (Original) The apparatus of claim 22 wherein said data frames are dropped at a fixed, predetermined rate.

24. (Original) The apparatus of claim 22 wherein said data frames are dropped at a variable rate.

25. (Currently Amended) The apparatus of claim 24 [[14]], wherein:

said processor is further for determining a communication channel latency;

said data frames are dropped at a decreased rate if said communication channel latency exceeds at least one predetermined threshold; and

said data frames are dropped at an increased rate if said communication channel latency falls below at least one other predetermined threshold.

26. (Original) The apparatus of claim 22, wherein said processor is further for determining a communication channel latency, for dropping said data frames at a first fixed rate if said communication channel latency falls below a predetermined threshold, and for dropping said data frames at a second fixed rate if said communication channel latency exceeds said predetermined threshold.

27. (Original) The apparatus of claim 22 wherein said processor is further for determining a communication channel latency, and for dropping each of said data frames having an encoded rate equal to a first encoding rate if said communication channel latency exceeds a predetermined threshold.

28. (Original) The apparatus of claim 27, wherein said processor is further for dropping each of said data frames having an encoded rate equal to said first encoding rate and a second encoding rate if said communication channel latency exceeds a second predetermined threshold.

a2

29. (Original) The apparatus of claim 22 wherein said means for generating data frames comprises:

a receiver for receiving a wireless communication signal; and

a demodulator for demodulating said wireless communication signal and for producing said data frames.

30. (Currently Amended) A receiver for achieving crypto-syncronization in a packet data communication system, the packet data communication system comprising a transmitter and said receiver, said transmitter and said receiver each having cryptographic security capabilities, said receiver comprising:

means for receiving ~~generating~~ data frames;

a queue for storing said data frames;

means for generating a state vector, said state vector incremented at a predetermined rate;

a decryption module for generating a codebook from at least said state vector, said codebook for decrypting at least one of said data frames; and

a processor for dropping one or more of said data frames in said queue and for adjusting said state vector for each of said data frames that are dropped.

31. (Original) The receiver of claim 30 wherein said processor adjusts said state vector by determining a number of dropped data frames and advancing said state vector in proportion to said number of dropped frames.

32. (Original) The receiver of claim 31 wherein said state vector is advanced by a value of one for each of said dropped data frames.

33. (Original) The receiver of claim 30 wherein said processor drops said one or more data frames at a fixed rate.

34. (Original) The receiver of claim 30 wherein said processor is further for determining a communication channel latency and dropping said one or more data frames at a variable rate in accordance with said communication channel latency.

35. (Original) The receiver of claim 34 wherein:

said processor decreases said rate if said communication channel latency falls below at least one predetermined threshold; and

said processor increases said rate if said communication channel latency exceeds at least one other predetermined threshold.

a2

36. (Original) The receiver of claim 30 wherein said processor is further for determining a communication channel latency; and

dropping said one or more data frames at a first predetermined fixed rate if said communication channel latency falls below a predetermined threshold; and

dropping said one or more data frames at a second predetermined fixed rate if said communication channel latency exceeds said predetermined threshold.

37. (Original) The receiver of claim 30 wherein said processor is further for determining a communication channel latency; and

dropping each of said one or more data frames having an encoded rate equal to a first encoding rate if said communication channel latency exceeds a predetermined threshold.

38. (Original) The receiver of claim 37 wherein said processor drops said one or more data frames having an encoded rate equal to said first encoding rate and a second encoding rate if said communication channel latency exceeds a second predetermined threshold.

39. (Original) A receiver for achieving crypto-syncronization in a packet data communication system, the packet data communication system comprising a transmitter and said receiver, said transmitter and said receiver each having cryptographic security capabilities, said receiver comprising:

means for generating data frames;

a queue for storing said data frames;

means for generating a state vector, said state vector incremented at a predetermined rate;

a2

a decryption module for generating a codebook from at least said state vector, said codebook for decrypting at least one of said data frames; and

a processor for disabling said state vector if no data frames are available to be decrypted in said queue.

40. (Original) The receiver of claim 39 wherein said state vector is enabled when at least one data frame becomes available for encryption in said queue.